TRIGRAMS'n'TAGS FOR LEXICAL KNOWLEDGE ACQUISITION

Berenike Litz, Hagen Langer and Rainer Malaka
TZI, University of Bremen, Germany

Keywords: Information extraction, Machine learning, Mining text and Semi-structured data, Lexical knowledge acquisition, Ontology learning, Ontology population.

Abstract: In this paper we propose a novel approach that combines syntactic and context information to identify lexical semantic relationships. We compiled semi-automatically and manually created training data and a test set for evaluation with the first sentences from the German version of Wikipedia. We trained the Trigrams'n'Tags Tagger by Brants (Brants, 2000) with a semantically enhanced tagset. The experiments showed that the cleanliness of the data is far more important than the amount of the same. Furthermore, it was shown that bootstrapping is a viable approach to ameliorate the results. Our approach outperformed the competitive lexico-syntactic patterns by 7% leading to an F_1-measure of .91.

1 INTRODUCTION

Lexical Knowledge Acquisition (LKA) is a highly relevant topic for many natural language applications. As the manual creation of their knowledge representation is a cost-intensive task, it is desirable to automatize this process as much as possible. Furthermore, the proposed representation sooner or later becomes obsolete and needs to be renewed or extended.

The idea to automatically acquire lexical knowledge dates back to 1981, when Amsler (Amsler, 1981) investigated the possibility to create a taxonomy from machine readable dictionaries (MRDs). He stated that the investigated machine-readable pocket dictionary offered a fundamentally consistent description of word meaning and in the future may provide the basis for research and applications in computational linguistic systems. The emergence of freely available online dictionaries such as Wikipedia[1] has improved the quality of lexical knowledge acquisition tremendously. While Snow et al. (Snow et al., 2005) achieved an F_1 value of only .36 on the task of hypernym extraction by using the Internet, Kazama and Torisawa (Kazama and Torisawa, 2007), who used Wikipedia, achieved a value of .88 with comparable methods.

In this work the data from the German version of the online encyclopedia Wikipedia was chosen for the task of hypernym extraction. This was done for the following reasons: Firstly, it contains up-to-date information, as it is editable by anyone. Therefore, e.g. named entities referring to persons who got famous recently are included. Secondly, it offers easy access to the whole encyclopedia and related data by a format convertible into a database. Thirdly, it is free to use, in contrast to other encyclopedias.

For the task of LKA lexico-syntactic patterns as proposed by Hearst (Hearst, 1992) are commonly used. Hearst patterns yield a much higher recall for information extraction from dictionaries such as Wikipedia than from the Web as a corpus. Furthermore, recent approaches have tried to automatically learn such patterns with the help of machine learning algorithms (Choi and Park, 2005; Etzioni et al., 2005). While these approaches perform considerably well, there is still enough headroom for increasing recall and precision as shown by our solution presented in the next paragraph.

2 OUR APPROACH

The Trigrams'n'Tags (TnT) Tagger by Brants (Brants, 2000) is known for its high accuracy and its multiplicity when it comes to new corpora, languages and tag sets. We employed the first sentences of the German

[1]URL: www.wikipedia.org (last access: 9th January 2009).

Litz B., Langer H. and Malaka R. (2009).
TRIGRAMS'N'TAGS FOR LEXICAL KNOWLEDGE ACQUISITION.
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, pages 18-25
DOI: 10.5220/0002292100180025

version of Wikipedia to train the tagger with a semantically enhanced tagset. By means of this approach we got hold of entities related by the hypernym/hyponym relationship.

The structured Wikipedia data promise already high F-measures for heuristic methods as shown by Kazama and Torisawa (Kazama and Torisawa, 2007) for the Japanese. Therefore, we implemented two baseline approaches, which we compared with our syntactic-semantic tagger. The first one applies the following notion: As the texts of encyclopedias follow some loose syntactic patterns, relevant information can be extracted with the support of heuristics about the syntactic-semantic distribution. This procedure is described in more detail in Section 4.

The other baseline implements an individual adjustment of the widely applied lexico-syntactic patterns by Hearst (Hearst, 1992). In comparison to completely unstructured text, the recall of the patterns is comparably high for encyclopedia entries. Section 5 presents the approach applying such patterns.

However, both heuristic methods have shortcomings. The simple employment of syntactic information is comparably low in precision whereas the lexico-syntactic patterns are rather low in recall. Our novel approach combines syntactic and context information to identify lexical semantic relationships. This can be exploited to determine hypernyms with very high precision and recall. Our method outperformed the highly competitive baselines by 7% leading to an F-measure of .91.

In the following sections the creation of the test and training data as well as the employment of the heuristic method, the lexico-syntactic patterns and the probabilistic model are described.

3 DATA PREPARATION

For all approaches, test data, which represent the gold standard need to be created. For the probabilistic model as described in Section 6, furthermore, training data had to be assembled, which consist of the first sentences of Wikipedia articles together with tags giving syntactic and semantic information.

Example 1, which was taken from Wikipedia, can illustrate this. The pattern **Hyponym** *is a* **Hypernym** can easily be detected.

(1) **Oliver Rolf Kahn** (* 15. Juni 1969 in Karlsruhe) *ist ein* deutscher **Fußballtorhüter** und derzeit in Diensten des FC Bayern München. (In English: **Oliver Rolf Kahn** (* 15th Juni 1969 in Karlsruhe) *is a* German **goalkeeper** and at the moment employed by FC Bayern München.)

3.1 Training Data

For supervised learning the training data is usually created by manually annotating a considerable amount of text. Therefore, obtaining training data by combining previously annotated data with manual work is of advantage.

Wikipedia articles include metadata for different named entity types, which can be applied for such a semi-automatic annotation. Especially interesting are so called *person data*, which are included in articles about persons to be automatically extracted and processed[2]. The data consists of fields such as *name*, *birthdate* and *-place*, *deathdate* and *-place* and a *short description* of the person. For the task of hypernym extraction the fields of *name* and *short description* were applied. A database was created, which includes these fields as well as the whole texts of the Wikipedia entries. For this work about 70,000 entries in the database were used for training and test.

The first paragraphs of the entries were annotated with the help of the part-of-speech (PoS) tagger Trigrams'n'Tags (TnT) by Brants (Brants, 2000) (for more details see Section 6) and with the tags PERSON (in case the token was the name of the person the article was about) and HYPERNYM (in case the token was a hypernym of PERSON), which replaced the PoS tags. For the semi-automatically created training set the tag HYPERNYM was given to part of the words of the sentences which were contained in the short descriptions (see Example (2)) as later described in more detail. All tags are part of the Stuttgart-Tuebingen-Tagset [3].

The final training file consisted of 291,388 tokens and experiments with smaller subsets showed that the biggest size slightly outperformed models with smaller data size, as shown in Table 1 in the Evaluation Section.

(2) Tamara_Ramsay/PERSON war/VAFIN eine/ART **Kinderbuchschriftstellerin./HYPERNYM** (In English: Tamara Ramsay was a **children's book author**.)

All training data was annotated with PoS tags. The SHORT DESCRIPTION (in the German template KURZBESCHREIBUNG) was used to extract

[2] For German: de.wikipedia.org/wiki/Hilfe:Personendaten (last access: 9th January 2009); For English: http://en.wikipedia.org/wiki/Wikipedia:Persondata (last access: 9th January 2009).

[3] see http://www.sfb441.uni-tuebingen.de/a5/codii/info-stts-en.xhtml (last access 19th January 2009).

hypernyms. As nouns are generally capitalized in German, all capitalized words were considered to be nouns and, therefore, hypernyms. The short description in Example 3 for the Spanish sailor Magellan could be utilized to extract *Seefahrer (sailor)* and *Krone (crown)*.

(3) portugiesischer **Seefahrer**, der für die spanische **Krone** segelte (In English: Portuguese sailor who sailed for the Spanish crown)

PoS tagging was not applied here, as the short descriptions are often only sentence fragments or single words and a tagger could not yield reliable results. However, in German, the selection of all capitalized words as nouns and named entities is nearly unfailing. The wrong annotation of non-hypernyms as e.g. *crown* in this example was considered a "repentance" to get hold of such a large amount of completely automatically annotated training data.

To sum up, for the different approaches, variations of the same training data were utilized. In all cases the untagged text were the first sentences of Wikipedia articles, however, the tagging was done with different tools. The only tag, that was given in advance was the tag PERSON for the named entities, which were the subject of the article.

- **Syntactic Information.** The creation of the training set for applying syntactic information was done with the help of a PoS tagger and the indication of nouns in German capitalized part of speech. In case there was a capitalized word, the PoS tag was replaced by the tag HYPERNYM.
- **Probabilistic Tagging with Semi-automatically created Training Data.** First, PoS tags were added by the same tagger and then the data base with the person data information was consulted. In all cases of capital words in the SHORT DESCRIPTION the PoS tag was replaced by the HYPERNYM tag.
- **Probabilistic Tagging with Manually annotated Training Data.** For this approach the semi-automatically created training data was proof-read by a person and in all wrong cases the HYPERNYM tag was replaced or inserted.

3.2 Test Data

For the creation of the test set a separate set with 4.000 tokens from the training data was corrected by an annotator and rechecked by another person. This data not only included the first sentences but also larger extracts from the Wikipedia texts, so that the gain and loss in precision according to the various approaches are apparent (see Example 4).

(4) Margaret Rutherford war eine britische **Schauspielerin**. Nachhaltige Berühmtheit erlangte sie in den frühen 1960er-Jahren durch die Darstellung der schrulligen Amateurdetektivin Miss Marple. Sie war mit Stringer Davis verheiratet.
(In English: Margaret Rutherford was a British **actress**. She became famous in the 1960s through her role as the detective Miss Marple. She was married to Stringer Davis.)

The untagged version of the test set was then tagged by each of the proposed methods and then compared with the tagged version, also referred to as the gold standard. All together the set contained 450 hypernyms.

The following two sections describe simple, straightforward applications, which were not in need of the presented training data. The supervised learning approach in Section 6, however, requires this data.

4 BASELINE 1: EMPLOYING SYNTACTIC INFORMATION

The first paragraphs of encyclopedia entries usually contain one or more hypernyms of the word the entry is about as well as few other nouns. Hence, it makes sense to take syntactic information about the words into account and to use this simple approach as a baseline. The most straightforward approach in this case is to tag any token marked as a noun (NN in the tagset) by a part-of-speech (POS) tagger as a hypernym. It can be expected that the recall of this approach is very high with a loss in precision. Example (2) shows a case where the approach performs well, Example (5) shows how the loss in precision can be explained: The sentences contain some nouns, which are not hypernyms at all.

(5) Neil_Ellwood_Peart/PERSON ist/VAFIN Texter/HYPERNYM und/KON Schlagzeuger/HYPERNYM der/ART **Rockband/HYPERNYM**
Rush./NE (In English: Neil Ellwood Peart is the songwriter and drummer of the **rockband** Rush.)

5 BASELINE 2: LEXICO-SYNTACTIC PATTERNS

A more sophisticated approach is the employment of lexico-syntactic patterns. Opposite to the previous method, this one is expected to be high in precision with a loss in recall.

Observing the data, it appeared that the two patterns in (2) and (3) are representable for most examples. For both patterns the typical constituents of a noun phrase (NP) are important (see Pattern (1)), which are article (ART), adjective (ADJA), noun (NN) and named entity (NE). Even though a hypernym is generally not a named entity, this tag was included in the pattern due to inaccuracy of the tagger.

$$NP = (ART) * (ADJA) * ((NN)|(NE)) \quad (1)$$

Pattern (2) takes the tag PERSON into account, which means, that the entry topic is part of the expression. The German verb forms *war, ist, waren* and *sind* all refer to the verb *sein* (in English: *be*).

$$\begin{array}{c}(PERSON)(war|ist|waren|sind)\\ (NP,) * (NP)((KON)(NP)) *\end{array} \quad (2)$$

The data of Wikipedia texts is only structured to some extend as it is mostly entered by laypersons. Therefore, a pattern disregarding the PERSON tag should also be tested, as the recall is likely to be higher (see Pattern 3).

$$\begin{array}{c}(war|ist|waren|sind)\\ (NP,) * (NP)((KON)(NP)) *\end{array} \quad (3)$$

Example 6, taken from the test data, can present this circumstance.

(6) Quirinus_Kuhlmann, **auch Culmannus, Kühlmann, Kuhlman,** war ein deutscher Schriftsteller. (In English: Quirinus_Kuhlmann, **also known as Culmannus, Kühlmann Kuhlman,** was a German writer.)

The sentence includes an insertion between the PERSON, the verb and the NP and, therefore, Pattern (2) will not be successful. However, the precision is likely to decrease with Pattern (3), as wrong sentences such as Example 7 are found.

(7) Ihre bekannteste Figur ist die blonde **Arzttochter** Annemarie Braun. (In English: Her best known character is the blond **doctor's daughter** Annemarie Braun.)

6 PROBABILISTIC TAGGING

For the creation of a statistical model we applied the Trigrams'n'Tags (TnT) Tagger by Thorsten Brants (Brants, 2000) trained with the NEGRA 2 corpus[4].

6.1 The Trigrams'n'Tags Model by T. Brants

TnT (Brants, 2000) utilizes second order Markov models (Rabiner, 1989), where the states represent tags and outputs represent the words. Transition probabilities depend on the states and the output probabilities only depend on the most recent category. For a given sequence of words $w_i...w_T$ of length T and the elements of the tagset $t_1...t_T$[5] Formula (4) is calculated. For a more detailed description see (Brants, 2000).

$$\underset{t_i...t_T}{\operatorname{argmax}} \left[\prod_{i=1}^{T} P(t_i|t_{i-1}, t_{i-2}) P(w_i|t_i) \right] P(t_{T+1}|t_T) \quad (4)$$

Furthermore, TnT applies linear interpolation for smoothing as trigram probabilities generated from a corpus usually cannot be used directly because of the sparse-data problem[6].

Unknown tokens are handled by taking into account the word ending, which is also important for hypernyms of persons, as those are often professions. For instance, a frequent ending for professions in German is *-er* (e.g. Maler, Musiker, Politiker; In English: painter, musician, politician) and the corresponding female word form of it with the ending *-erin* (correspondingly, Malerin, Musikerin, Politikerin). In Table 1 (described in more detail in Section 7.3.1) the necessity of such a method can be seen. The higher percentage of unknown tokens for smaller data sets, which was 42% for 1.25% of the training data in comparison to 19% for 100%, only slightly decreased the accuracy in our training data. There, the value varied between 0.936 and 0.948 for the corresponding percentages. The optimal length of the suffix tries was evaluated as presented in Section 7.3.3.

[4]The NEGRA corpus version 2 consists of 355,096 tokens (20,602 sentences) of German newspaper text, taken from the Frankfurter Rundschau. For more information visit: http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html (last access: 9th January 2009).

[5]the additional tags t_{-1}, t_0, and t_{T+1} are beginning-of-sequence and end-of-sequence markers.

[6]This means there are not enough instances for each trigram to reliably estimate the probability.

During the parameter generation step, the lexicon and an n-gram file are created by TnT. These files contain the frequencies of tokens and tags in the training data and are needed for the prediction of the probability for a specific tag at a specific location. The lexicon file contains the frequencies of tokens and their tags as they occurred in the training data. These frequencies are needed to determine lexical probabilities in the tagging process. The n-gram file contains the contextual frequencies for uni-, bi- and trigrams.

With the help of the lexicon and the n-gram file, predictions about the probability of a particular tag can be made.

7 EVALUATION AND RESULTS

The quality of the results was measured by the values *precision* (P) and *recall* (R) and combined by the F_1-*measure* (F_1) (Van Rijsbergen, 1979). Furthermore, the *accuracy* (A) was calculated.

7.1 Syntactic Information

Figure 1 shows the results that were calculated for the syntactic information approach. For the PoS tagging we compared two different taggers: The Trigrams'n'Tags (TnT) Tagger by Thorsten Brants[7] (Brants, 2000) and qtag by Oliver Mason (Tufis and Mason, 1998). As shown in Figure 1, with an F_1 of .55 the results of Tnt are much more promising than the one of qtag with .47 for the approach taking only nouns (NNs) into account.

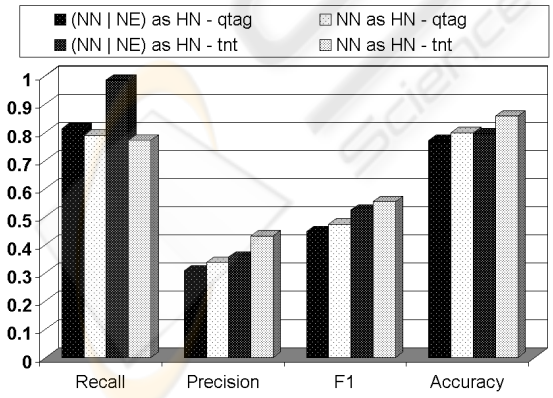

Figure 1: A comparison of four different possibilities for applying syntactic information for lexical knowledge acquisition: All nouns ((NN); and named entities (NE)) are tagged as hypernyms with the taggers qtag and TnT.

[7]For a description of the TnT algorithm see Section 6.

The recall of the best method was not near 100 % (as can be seen in column 4 of Figure 1) because in many cases nouns (NN) were tagged by the POS tagger as named entities (NE). Once the named entities are replaced by the HYPERNYM tag the recall yielded 97 % with a further loss in precision as shown in the first two columns. The loss in precision gave reason to the resulting lower F_1 of the approach including named entities. Therefore, the approach taking only nouns into account outperformed this one by .03 for qtag as well as for TnT.

7.2 Lexico-Syntactic Patterns

The lexico-syntactic patterns presented in (2) and (3) were applied to the test data and compared with the gold standard. Figure 2 shows the result of the two patterns, which were tested with and without taking NEs into account as hypernyms. The precision of Pattern (2) which only takes NNs into account, was highest. However, highest overall results with an F_1 value of 0.85 were achieved for Pattern (3) including NEs. The diagram depicts the similarity between the precision values of the patterns and the comparatively higher F_1-value of the pattern including NEs but excluding PERSON.

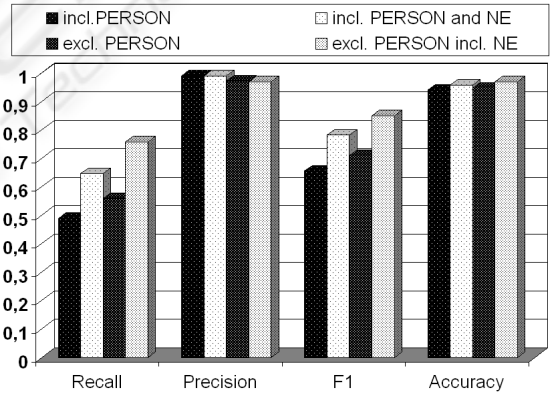

Figure 2: A comparison of lexico-syntactic patterns for lexical knowledge acquisition from Wikipedia.

7.3 Semantic Tagging

The evaluation of the approaches with the probabilistic training data is presented in the following. As a first step we evaluated the performance of the semi-automatically created model as presented in Section 7.3.1. Afterwards, we compared the results with a small amount of manually annotated training data in Section 7.3.2. In Section 7.3.3 we calculated the best suffix length for the suffix tries for unknown words. The best results for linear interpolation are presented

Table 1: Results of the training data and percentage of Unknown Tokens (UT) with respect to the corpus size measured with Recall (R), Precision (P), F_1, Accuracy (A).

	1.25%	**12.5%**	**25%**	**50%**	**100%**
R	0.767	0.802	0.818	0.807	0.802
P	0.696	0.722	0.72	0.742	0.75
F_1	0.729	0.76	0.766	0.773	**0.776**
A	0.936	0.943	0.944	0.947	**0.948**
UT	42.13	27.59	24.89	21.04	18.90

Table 2: Results of percentages of manually annotated training data with respect to the whole semi-automatically created training data measured with Recall (R), Precision (P), F_1, Accuracy (A).

	1.21%	**2.49%**	**3.17%**	**4.16%**
R	0.919	0.908	0.895	0.917
P	0.792	0.831	0.844	0.847
F_1	0.851	0.868	0.869	**0.881**
A	0.963	0.968	0.969	0.972

in Section 7.3.4. These results and the ones from the suffix trie were taken into account for the bootstrapping method in Section 7.3.5.

7.3.1 Semi-automatically Created Training Data

First experiments with the semi-automatically created training data showed that the total size of around 300,000 tokens played only an inferior role for the quality of the results (see Table 1). Even 1.25% of the data yielded promising results. Therefore, even considerably smaller data sets can be taken into account for this approach. This is particularly interesting for approaches, where no annotated data is available. Even though the number of unknown tokens (UT, in Table 1) increases with a decrease in training data, the tagger performs competitively.

7.3.2 Manually Annotated Training Data

The findings about the data size described in the previous paragraph led to the idea of manually annotating a small amount of training data. The results in Table 2 show that the quality increases slightly but steadily with the size of manually annotated data.

As the 4.16% model outperformed the other ones, it was taken for the experiments with the suffix length, the linear interpolation and the bootstrapping.

Table 3: Results of percentages of manually annotated training data of the evaluation of suffix length for four percent manually annotated training data. Suffix length (SL) 0 to 7 measured with Recall (R), Precision (P), F_1, Accuracy (A).

SL	**R**	**P**	F_1	**A**
0	0.606	0.946	0.738	0.951
1	0.861	0.771	0.814	0.955
2	0.913	0.831	0.870	0.969
3	0.922	0.838	0.878	0.971
4	0.930	0.844	**0.885**	0.972
5	0.928	0.845	**0.885**	0.972
6	0.924	0.846	0.883	0.972
7	0.922	0.846	0.882	0.972

7.3.3 Suffix Length

The word ending is an indication for the presumptive part of speech of an unknown word in a corpus. For the task of hypernym tagging this is also applicable. We conducted experiments with varying suffix length to see, if the default of length=10 was the best choice for the task.

The results in Table 3 show that the suffix lengths of 4 and 5 were nearly the same for F_1, and that they both outperformed the default.

7.3.4 Linear Interpolation

Due to the sparse-data problem, trigram probabilities generated from a corpus usually cannot directly be used. TnT uses linear interpolation of unigrams, bigrams, and trigrams, as this smoothing paradigm here delivers the best results. The trigram probability can be estimated with Formula (5), where $\hat{P}$ are maximum likelihood estimates of the probabilities. The sum of λ_1, λ_2 and λ_3 is 1 as P represents probability distributions.

$$P(t_3|t_l,t_2) = \lambda_1\hat{P}(t_3) + \lambda_2\hat{P}(t_3|t_2) + \lambda_3\hat{P}(t_3|t_1,t_2) \quad (5)$$

The default setting for TnT is computed with deleted interpolation. This is an individual adjustment of the values according to the model. In the case of our model the values $\lambda_1 = 0.1629941$, $\lambda_2 = 0.2463481$ and $\lambda_3 = 0.5906578$ were calculated by this algorithm.

We evaluated an adjustment of the λ values for our task and it appeared that the values $\lambda_1 = 0.1$ and $\lambda_2 = 0.0$ yielded only slightly better results (from .88075316 to 0.88272923 for F_1) as shown in Figure 3. However, even if the improvement was marginal, for a bootstrapping of the approach it made a difference as shown in the following section.

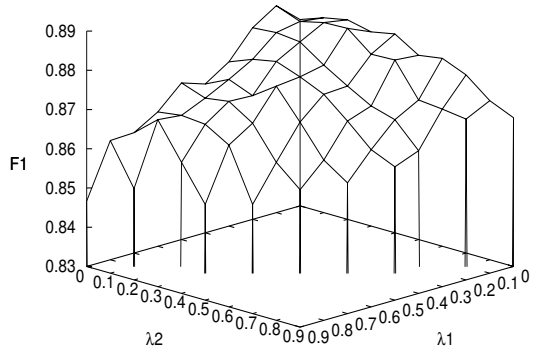

Figure 3: F_1 results for all possible λ_1 and λ_2 values.

7.3.5 Bootstrapping of Training Data

The increasing F_1-values for an increasing amount of manually annotated training data gave reason to conduct experiments compensating lacking training data. However, the first experiments with semi-automatically created data were not successful. Therefore, an approach needed to be chosen which, on the one hand, makes use of the manually created training data, and, on the other hand, can countervail cost-intensive annotation. These points led to the conclusion to perform a bootstrapping approach as it is interpreted by Abney (Abney, 2002): " a problem setting in which one is given a small set of labeled data and a large set of unlabeled data, and the task is to induce a classifier."

The experiment was accomplished with the following steps:
(1) Split unlabeled data into files containing 5000 tokens.
(2) Take manually created trainset to tag first unlabeled data file.
(3) Concatenate the manually labeled data and the newly tagged file.
(4) Create a new model with the concatenated file.
(5) Continue with Step (2) till all split files are tagged.

The findings of the best choice of suffix length and for the λs for linear interpolation were utilized to boost the results of bootstrapping.

Figure 4 shows the values of the three bootstrapping approaches: the default one; the one considering the best λ values; and the one considering the best λ values plus the best suffix length. For the default it appeared that after a considerable increase between the inclusion of 26 and 28 files a maximum of F_1=.904 was reached.

For the results including the best choices of linear interpolation with $\lambda_1 = 0.1$ and $\lambda_2 = 0.0$ (see Figure 3), the best model was created with 40 and 41 bootstrapping files and the F_1 measure yielded a value of nearly .9123.

As the results of the suffix lengths 4 and 5 were nearly the same, they were both evaluated for the bootstrapping. Here, it turned out that the suffix length=5 performed better. The best results for F_1 of suffix length=4 were .90831554 for (48 files) and for length=5 it was .91257995 (52 files). These were the highest results measured for all evaluations.

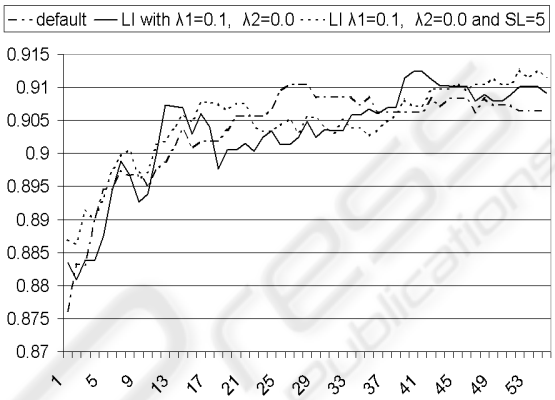

Figure 4: Bootstrapping results for F_1 with varying file number for the default, for linear interpolation with $\lambda_1 = 0.1$ and $\lambda_2 = 0.0$ and for linear interpolation with $\lambda_1 = 0.1$ and $\lambda_2 = 0.0$ including suffix length=5.

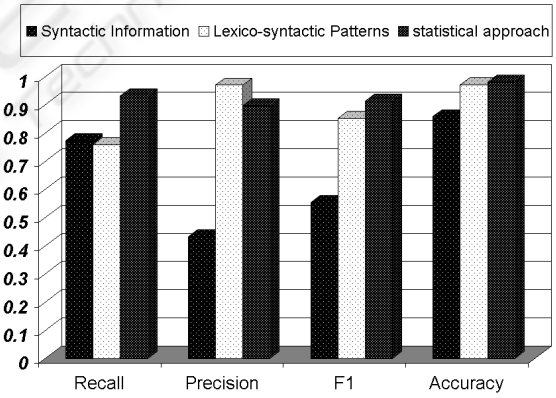

Figure 5: Classification with Dictionaries: Comparison of the Syntactic Information, the Lexico-Syntactic Patterns and the Statistical Model Approach.

7.4 Comparison of Approaches

Figure 5 shows a comparison of the results created by the best statistical model and the syntactic and lexico-syntactic pattern approaches. The syntactic information baseline outperformed the lexico-syntactic patterns slightly with the recall value but was inferior in all other aspects. The lexico-syntactic patterns appeared competitive with the statistical approach.

8 CONCLUSIONS

In this paper we presented an approach, which yielded outstanding results for lexical knowledge acquisition. Even though the lexico-syntactic patterns performed competitively with the first test runs of the statistical approach, some adjustments could outperform them. The first run utilized a semi-automatically created training data set containing 300.000 tokens, which resulted in $F_1 \approx .78$. It appeared that a fractional amount of the data (around 4%), which was manually corrected, outperformed those results with $F_1 \approx .89$. A suffix length=5 and an adjustment of the λ values for linear interpolation only gave slight improvements on the third position of the decimal point. However, for the bootstrapping appoach these minimal improvements became apparent and resulted in an F_1 value of over .91.

Future Work involves the deployment of other statistical models to the given data. One choice is to apply conditional random fields (Lafferty et al., 2001). Here, the evaluation of the performance of Hidden Markov Models and other statistical models on an unseen domain is an important step towards generalization. The transfer of the model (initially adjusted to persons) to e.g. general processes or things will not be a challenge as the first sentences in Wikipedia are mostly identically structured.

ACKNOWLEDGEMENTS

This research was only possible with the financial support of the Klaus Tschira Foundation and the CONTENTUS Use Case of the THESEUS Program funded by the German Federal Ministry of Economics and Technology (BMWi).

REFERENCES

Abney, S. (2002). Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 360–367, Morristown, NJ, USA. Association for Computational Linguistics.

Amsler, R. A. (1981). A taxonomy for english nouns and verbs. In *Proceedings of the 19th Annual Meeting on Association for Computational Linguistics*, pages 133–138, Morristown, NJ, USA. Association for Computational Linguistics.

Brants, T. (2000). TnT – A statistical Part-of-Speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing (ANLP-2000)*, pages 224–231, Seattle, Washington.

Choi, S. and Park, H. R. (2005). Finding taxonomical relation from an mrd for thesaurus extension. In Dale, R., Wong, K.-F., Su, J., and Kwong, O. Y., editors, *Natural Language Processing - IJCNLP*, volume 3651 of *Lecture Notes in Computer Science*, pages 357–365. Springer.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165(1):91–134.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING*, Nantes, France.

Kazama, J. and Torisawa, K. (2007). Exploiting wikipedia as external knowledge for named entity recognition. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML-01*.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77.

Snow, R., Jurafsky, D., and Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press, Cambridge, MA.

Tufis, D. and Mason, O. (1998). Tagging romanian texts: a case study for qtag, a language independent probabilistic tagger. In *Proceedings of the 1st International Conference of Language Resources and Evaluation (LREC-98)*, Granada, Spain.

Van Rijsbergen, C. J. K. (1979). *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow.